

ASE 230 Tools Installation - PHP

Recommendations

Recommendation for Windows Users

- **Start with XAMPP** if you are not familiar with setups and configurations for ease and rapid development.
- **Move to manual installation** to gain practical, professional-level experience.
- This progression ensures both **accessibility** and **technical depth** for students.

Linux is a must for web applications

- **Stuents must learn how to use Linux**
 - Most web applications are running on Linux.
 - Windows support Linux with WSL2.
- We will use **Docker**, and it uses Linux for web application development.
 - Docker on Windows are slow and error prone.
 - **Students are required to use *Docker*** for many ASE course projects.

Recommendation for Mac/Linux Users

- **Start using manual installation** as it is easy to install PHP/MySQL on Mac/Linux with CLI (command line interface) tools.
- **Use homebrew**
 - Install Homebrew if not installed:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

PHP Setup and Installation

Development Environment Setup

Option 1: XAMPP/MAMP (Recommended only for beginners)

```
# Download XAMPP from https://www.apachefriends.org/  
# Includes Apache, MySQL, PHP in one package  
# Start Apache from XAMPP control panel  
# PHP is automatically configured
```

Option 2: Standalone PHP

```
# macOS with Homebrew
brew install php
php -v # Check PHP version

# Ubuntu/Debian
sudo apt update
sudo apt install php php-cli php-mysql
php -v # Check PHP version
```

Windows installaton

1. Open <https://windows.php.net/download/>
2. Choose the Zip file and download in your computer.
3. Unzip to your php directory (for example, c:\bin\php)
4. Add PHP to system PATH.
5. In the php directory, copy `php.ini-development` to `php.ini`.

6. Find these lines and remove comment (the first ; character).

```
extension=mysqli
...
extension=pdo_mysql
...
extension_dir = "ext"
```


Option 3: Docker (Advanced)

```
# Run PHP with Apache
docker run -d -p 8080:80 --name php-apache php:8.2-apache

# Run PHP CLI only
docker run -it --rm php:8.2-cli php -v
```

Verify PHP Installation

Check PHP Version

```
php -v
```

Test PHP with Web Server

```
# Start built-in PHP development server  
php -S localhost:8000  
  
# Create test file  
echo "<?php phpinfo(); ?>" > info.php  
# Visit http://localhost:8000/info.php
```

Essential PHP Extensions (optional)

Common Extensions for Web Development

```
# Ubuntu/Debian
sudo apt install php-curl php-json php-mbstring php-xml php-zip

# macOS with Homebrew
brew install php@8.2
# Extensions usually included

# Windows
# Enable in php.ini file:
# extension=curl
# extension=mbstring
# extension=openssl
```

PHP Configuration

Important php.ini Settings

```
; Display errors for development
display_errors = On
error_reporting = E_ALL

; File upload settings
upload_max_filesize = 10M
post_max_size = 10M

; Memory limit
memory_limit = 128M

; Session settings
session.save_path = "/tmp"
```

Composer Installation

PHP Package Manager

```
# Download and install Composer globally
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer

# Windows (using installer)
# Download from https://getcomposer.org/download/

# Verify installation
composer --version
```

Testing Your Setup

Create a Simple PHP Script

```
<?php
// test.php
echo "Hello, PHP World!\n";
echo "PHP Version: " . phpversion() . "\n";
echo "Current Date: " . date('Y-m-d H:i:s') . "\n";

// Test database connection (if MySQL available)
if (extension_loaded('mysqli')) {
    echo "MySQL extension is loaded!\n";
} else {
    echo "MySQL extension not found.\n";
}
?>
```

Run the Test Script

```
# From command line  
php test.php
```

```
# From web browser (if web server running)  
# Place test.php in web root directory  
# Visit http://localhost/test.php
```

IDE and Editor Setup

Recommended Development Tools

- **VS Code** with PHP extensions
- **PHPStorm** (Professional IDE)

VS Code Extensions

- PHP IntelliSense
- PHP Debug
- PHP DocBlocker
- Prettier – Code formatter

Troubleshooting

Common Issues and Solutions

PHP command not found

```
# Add PHP to PATH (macOS/Linux)
export PATH="/usr/local/bin:$PATH"

# Windows: Add PHP folder to system PATH
```

Extension not loading

```
# Check loaded extensions
php -m

# Check php.ini location
php --ini
```

Resources

Learning and Documentation

- Official PHP Documentation:
<https://www.php.net/docs.php>
- PHP: The Right Way: <https://phptherightway.com/>
- W3Schools PHP Tutorial:
<https://www.w3schools.com/php/>
- Laracasts: <https://laracasts.com/> (Premium tutorials)

Practice Platforms

- PHP Playground: <https://onecompiler.com/php>
- 3v4l.net: PHP version comparison tool